Summary of SEI Accomplishments: 2Q91

This quarter, the **Software Capability Evaluation** course was taught at the Air Force Systems Command/Electronic Systems Division in Boston.

In April, Software Capability Evaluation Project members formed the **Software Capability Maturity Model Project**. This project will maintain an organizational model of software process maturity.

The **Software Metrics Definition Working Group** and the **Software Acquisition Metrics Working Group** met for three days each. They agreed to accelerate their schedules to have reports ready for distribution at the August 1991 SEI Affiliates Symposium.

The **Software Process Definition Project** held a three-day workshop at Standard Systems Center entitled **"Defining the Software Process – Getting Started."**

Software Architectures Engineering Project members developed and delivered a course for training Naval Coastal Systems Center software developers in the adaptation and use of the Object Connection-Update structural model.

The **CASE Technology Project** hosted a workshop on CASE management issues in June. Representatives from industry and government participated in the exchange.

The Next Generation Computer Resources has selected IEEE Futurebus+ as its computer backplane standard. The **Rate Monotonic Analysis for Real-Time Systems Project** has worked to ensure that rate monotonic scheduling is the recommended approach for real-time applications using Futurebus+.

The **Real-Time Embedded Systems Testbed (REST) Project** developed a half-day tutorial on evaluation and benchmarking that summarizes years of experience of the REST Project. The tutorial was offered at the Washington Ada Symposium.

The **final version** of Serpent was released. An announcement about the opportunity to license Serpent from CMU will appear in *Commerce Business Daily* in early July.

The **Systems Fault Tolerance Project** hosted a Fault Tolerant Systems Practitioners Workshop at the SEI in June.

The **Software Risk Management Program** developed a questionnaire for the government (derived from the Risk industry questionnaire) that will gather information on how DoD services handle risk management and risk communication. The questionnaire was used in Risk's first government program office interview in May 1991.

The **Continuing Education Project** presented two executive seminars from the Continuing Education Series: "Software: Profit through Process Improvement" and "Software Quality Improvement."

New **affiliates** from five industry organizations and seven academic institutions signed information exchange agreements during this quarter. New **resident affiliates,** one from Computer Sciences Corporation and one from the DoD, recently joined the SEI.

This section provides a summary of accomplishments from April–June 1991

Accesion For	
NTIS CRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	

By
Dist ibution /

Availability Codes	
Dist	Avail and / or Special
A-1	

Table of Contents

Software Process

The Software Process Program focuses on improving the process of software development. Projects within the program are: assessing the actual practice of software engineering in the defense community, training organizations to gain management control over their software development processes, supporting the use of quantitative methods and measures as a basis for process improvement, and developing improved methods for software process management.

The Software Process Assessment (SPA) Project helps organizations improve their software development process by providing a structured framework for assessing their current practice. The objectives of an assessment are to determine the organization's software process maturity, identify key areas for improvement, and help the organization initiate those improvements.

● ● ● ● ● ● ● ● ● ● ● ● ● ●
Software Process Assessment

The SPA Project provides assessment briefings, training, and coaching services. In addition, it supports a set of nine commercial assessment vendors. The project is currently focusing its efforts on assessments as components of comprehensive process improvement programs.

In the second quarter, SPA Project members conducted two software process assessment training classes. The first class was for the Air Force Logistics Command. The second class included Hewlett Packard, IBM, Army Material Command, GTE Government Systems, and Naval Air Systems Command. In addition to these training classes, project members also conducted an Assessment Tutorial at the Washington Ada Symposium and a Software Process Assessment Briefing at the SEI.

During this quarter, SPA staff members provided SEI assessment coaches for assessments conducted by Science Applications International Corporation and Motorola Cellular Infrastructure Group.

Also during this quarter, SPA Project members participated in the ongoing development of the Software Capability Maturity Model.

The Software Capability Evaluation (SCE) Project helps Department of Defense (DoD) acquisition organizations evaluate the capability of contractors to develop and maintain software competently. The project is improving and implementing an evaluation method that examines the software process of contractors for use in software-intensive acquisitions.

● ● ● ● ● ● ● ● ● ● ● ● ● ●
Software Capability Evaluation

During this quarter, Richard Kauffold became the SCE Project Manager. Also in this quarter, project members trained evaluation teams from the Air Force Logistics Command, Naval Air Development Center (NADC), and the United States Army. Project members presented an overview of the SCE method and led

the teams through simulated visits to two contractor sites. The training course was held at the SEI, where five individual teams were trained.

During May, the SCE course was taught exclusively for the Air Force by an SCE project member at the Air Force Systems Command/Electronic Systems Division (ESD) in Boston. The 3-day course was attended by 24 people and merited positive evaluations.

In May, project members visited the Defense Systems Management College (DSMC) to deliver the SCE overview course and conduct distribution partnership discussions.

The Software Process Maturity course was given in June as part of the DSMC's Advanced Software Management Course. Also in June, project members trained teams from the Defense Systems Management College, NADC, Naval Ocean Systems Center, Naval Training Systems Center, and the Army Program Manager, Training Devices Command.

●●●●●●●●●●●●●
Software Capability Maturity Model

The Software Capability Maturity Model (SCMM) Project maintains an organizational model of software process maturity. This model will be continuously updated with the state of the art as it evolves in software engineering, total quality management, and other relevant areas of improvement. It will elaborate software development practices that provide clear strategies for capability maturity growth and improvement.

The SCMM Project grew out of the SCE Project. In April, SCE Project members responsible for maintaining and improving the Software Capability Maturity Model and Maturity Questionnaire formed the SCMM. During this quarter, SCMM project members, along with members from other Software Process Program projects, continued work on documents that will be released at the SEI Affiliates Symposium in August.

One document is a paper describing the Software Capability Maturity Model and how it is used in process assessments and capability evaluations. Another is a list of software management and engineering practices that in most cases should be implemented at each maturity level. This document will serve as a guide for software professionals wishing to improve the maturity of their organizations' software processes. It will also serve as a reference for assessment and evaluation teams when conducting on-site investigations of an organization's software process.

The list of practices was first released last summer for public comment. It was revised during the fall and winter and again distributed for public review this quarter. In April, the list of practices for Maturity Level 2 was distributed to approximately 150 working group members from 88 companies for review. The Level 3 practices were mailed in May, and Levels 4 and 5 were mailed on July 1.

Using the practices and questions from the preliminary version of the Maturity Questionnaire (CMU/SEI-87-TR-23), project members began generating questions for a revised Maturity Questionnaire. Attendees at the Affiliates Symposium will have an opportunity to work with a draft version.

The Software Process Measurement Project advocates the use of measurement in managing software development. The project is formulating reliable measures of the software development process and products to guide and evaluate development. To expedite DoD and industry acceptance, the project convenes a steering committee and two working groups, and undertakes a best-practices study, all of which are devoted to encouraging organizations to use quantitative methods to improve their software processes.

During this quarter, the Software Metrics Definition Working Group—composed of the Quality, Size, and Effort/Schedule subgroups—and the Software Acquisition Metrics Working Group met for three days each. The working groups agreed to accelerate their schedules to have reports ready for distribution at the August 1991 SEI Affiliates Symposium. Comments resulting from this broad external review will be incorporated into final technical reports, which are targeted for release in the first quarter of 1992. During this quarter, two revisions have been completed for each of the four reports listed below:

- Software Effort and Schedule Measurement
- Software Quality Measurement: Problem Reporting and Tracking
- Software Size Measurement
- *Software Measurement Guide for Acquisition Program Managers*

Project members reviewed the final draft of the technical report *Measurement in Practice*. This report will be available for distribution at the 1991 SEI Affiliates Symposium.

Anita Carleton and Robert Park prepared task descriptions of the Software Process Measurement and Software Cost Estimation tasks. They presented overviews to Bob Webster and Chuck Shank in support of the Computer Resource Management Technology Program Task Review.

Members of the project continued working with Charles Koch, of NADC, to provide assistance in assembling a measurement implementation plan. In May, Robert Park attended the annual conference of the International Society of Parametric Analysts, and Anita Carleton attended a "Total Service Quality" course offered by the Juran Institute. Anita Carleton also participated in the SEI CASE Management Workshop where one of the sessions specifically dealt with discussions of CASE and Metrics. Project members visited the Unisys Corporation in Houston, Texas, to study defect data and to discuss piloting some of the project definitional work.

Several project members participated in an information exchange with the Deputy Assistant Secretary of Defense for Production Resources, Nicholas Torelli, Jr., at the Pentagon.

Project members met with Col. Ron Bowen, Maj. Keith Bowman, and Brad Donald at the Air Force Cost Analysis Center to outline the SEI's plans for cost management improvement activities and to explore ways in which the SEI could collaborate with the cost center to improve cost estimation.

Robert Park met with Raytheon representatives to explore ways in which Raytheon and the SEI could work together in software measurement.

●●●●●●●●●●●●●●●
**Software Process
Definition**

The objectives of the Software Process Definition Project are to establish the use of defined processes for the management and development of software as standard software engineering practice and to advance the capabilities required to define the software process within an organization. A "defined process" means that a process is documented, supported by training, and practiced, and that the practice, training, and documentation are generally found to be equivalent.

The project is supporting process development efforts at Standard Systems Center (SSC), Gunter Air Force Base. Also, the project explores advanced applications of process through the Software Technology for Adaptable, Reliable Systems (STARS) program.

During this quarter, the project held a three-day workshop at SSC entitled "Defining the Software Process – Getting Started." The workshop covered basic principles of process management, process improvement, process definition, and the descriptive modeling process. The workshop also included group exercises that allowed the participants to apply what they had learned. The project provided on-site engineering support to SSC process improvement working groups in the development of working group products, including:

- Software Configuration Management Working Group Tactical Plan

- Management Oversight Working Group Charter

- Project Management Working Group project management bibliography

- Project Management Working Group descriptive process model of current project management practice

In addition, project members presented a summary of accomplishments to senior management and SSC personnel involved in the SSC improvement effort. During this period, project members developed a joint SSC/SEI presentation for the 1991 SEI Affiliates Symposium entitled "Software Process Definition: Principles to Practice." Project members prepared and conducted an on-site review of the SSC technical objectives and plans (TO&P) and project support plan for the coming fiscal year.

Project members continued to provide support for the STARS Program. Project members serve on the STARS Process Joint Activities Group as an FFRDC facilitator. In this capacity, project members assisted the process component of the STARS effort by participating in:

- The development of strategies, concepts, and plans for guidance and direction.

- The planning and development of an operational concept document.

The "Process Operational Concept Document" contains the technical foundation and descriptions of the operational capabilities proposed for the STARS environment. Project members help review and assess the STARS contractor T-increment proposals.

A project member serves on the Air Force Process Working Group as deputy chair. During this quarter, the project member participated in a technical information exchange and review of technology that is applicable to the process component of the STARS effort. The project also hosted an Air Force Process Working Group meeting.

Project members represented STARS at the Defense Advanced Research Program Agency (DARPA) Software Process Workshop. Additionally, they provided logistics support, planning, and coordination for this workshop.

As participants in the development of STARS technology, project members began the analysis and evaluation of experience-tested process models collected during the past two quarters. Project members began planning for the next STARS advisory committee meeting. This task will lead to the development of a repository of reusable process assets that will support the STARS environment. This task will be accomplished in conjunction with STARS prime contractors.

Project members also participated in and made presentations at technical information meetings with General Electric, GTE, and Raytheon.

The purpose of the Software Acquisition Process Development Project is to improve the state of the practice in software acquisition. The project will adapt and tailor existing software acquisition processes and, where necessary, create new process definitions.

● ● ● ● ● ● ● ● ● ● ● ● ● ●
Software Acquisition Process Development

During this quarter, project members focused on completing and piloting the Software Proposal Evaluation Practices Questionnaire. This was the first step in the plan to develop a methodology for evaluating contractors' proposals for implementing software. To date the questionnaire has been used to solicit data from five projects at the Air Force Systems Command/Electronic Systems Division at Hanscom Air Force Base. Additional projects from other Air Force product divisions, as well as CECOM and the Naval Air Systems Command, will contribute data to the survey. The survey will continue through the third quarter, with data analysis and a final report to follow by the end of the year.

Software Engineering Techniques

The primary objective of the Software Engineering Techniques Program is to improve the practice of software engineering by improving individual and team productivity through the identification and transition to practice of emerging software technology. Promoting the appropriate use of this technology supports the SEI effort to transform software development from an ad-hoc, labor-intensive activity to a technology-supported engineering discipline.

To develop a fundamental understanding of structures for the software architecture level of design, the Software Architecture Design Principles Project is describing basic design elements used in the description, analysis, and development of software systems.

Software Architecture Design Principles

Project members visited Apple Technology Group (Cambridge, Massachusetts), Massachusetts Institute of Technology (Cambridge, Massachusetts), and the University of Massachusetts (Amherst, Massachusetts) to present and discuss material on software architectures.

The project leader presented a paper on software engineering education at a workshop in conjunction with the International Conference on Software Engineering (ICSE), held in Austin, and accepted an award for "the most influential paper from ICSE-2." In her acceptance speech, the project leader reviewed the history of abstract data types.

The following project papers were accepted for publication in this quarter:

- "Heterogeneous Design Idioms for Software Architecture," for the Sixth International Workshop on Software Specification and Design.

- "Models for Undergraduate Project Courses in Software Engineering," for the Fifth SEI Conference on Software Engineering Education.

Public service activities performed during this quarter included participation in meetings with the Nuclear Regulatory Commission Computer Science and Telecommunications Board, DARPA Information Science and Technology Study Group, and the Advisory Board for Office of Technology Assessment's *Intellectual Property Project*.

The Software Architectures Engineering (SAE) Project has been involved in the engineering of software in several critical DoD application areas for the past five years. The project goal is to use sets of identified patterns together with engineering design goals to provide guidelines for building the basic elements of software structure (models).

Software Architectures Engineering

The project is currently focusing on three aspects of this technology:

- The generalization of software architecture by providing templated structural elements and connection rules that invest the implemented software with the desired product qualities.

- The naturalization of the models so that real-world components have corresponding software analogs.

- The application of tools and notations to software development based on generalized structures.

The SAE Project provides DoD program offices with improvements to the practice of software engineering by assisting in the creation and adoption of structural, model-based technology. The project accomplishes these improvements by helping clients to abstract the desirable software architecture characteristics for particular classes of applications and to use them to create reusable patterns of software structures specific to an application. Project members refine and mature the new structural models by transitioning them to other projects, which allows additional sources of reflection on their use.

The SAE Project completed the first phase of work with the Air Force ~lectronic Combat Office (AFECO) in conjunction with the CROSSBOW-S organization. Project members began converting the AFECO report, delivered last quarter, into an SEI technical report.

Project members continued working with the CROSSBOW-S Digital Simulation Steering Group Architectures Working Group, charged with producing a designer's handbook for a standard electronic combat test and evaluation system. SAE project members are serving as reviewers and as a source for handbook material.

Project members are also working with the DARPA Domain-Specific Software Architectures (DSSA) Program. This research program will fund several industrial and academic teams to focus on software architecture development in DoD critical domains. The SAE Project is refining a plan for Lt. Col. Erik Mettala, Manager of the DARPA DSSA Program, outlining SAE project tasks in the continuing support of the DARPA DSSA effort. SAE project members met with members of the SEI Domain Analysis Project to determine a set of activities for working jointly with the research teams.

Finally, SAE project members are continuing to work with the Naval Coastal Systems Center (NCSC) on the AN/SSQ-94 Combat System Integrated Training Equipment (CSITE) Program. Together with a resident affiliate from NCSC, who joined the SAE Project in December, project members developed and delivered a course for training NCSC software developers in the adaptation and use of the Object-Connection-Update structural model. An NCSC design/architecture working group was formed and has begun meeting with SAE project members. One goal of the group is to formulate and document the structural model to be used for CSITE for review in August.

The Software Development Environments (SDE) Project provides environment support for software configuration management (SCM). In particular, the project has been assessing commercial advances in SCM concepts and functionality and is investigating issues of integration among SCM, development tools, and environments.

Configuration management capabilities can be found in SCM tools, computer-aided software engineering (CASE) tools, and environment frameworks, each implementing its own variant of some of the concepts. This variety leads to the need to integrate tools with different SCM capabilities into a development environment, and the desire for a unified SCM model that can be easily adapted. As a result, the project has focused on the following activities:

- Creating a framework for relating SCM concepts found in actual SCM systems, including four conceptual models, and assessing their impact on the software process.

- Providing an understanding of SCM as a key component of an SDE infrastructure.

- Addressing issues of integration between tools and SCM capabilities in environment frameworks with diverse SCM concepts.

- Exploring the feasibility of a unified SCM model and interface.

This quarter, *Proceedings of the 3rd International Software Configuration Management Workshop (SCM3)*, Peter Feiler (Ed.), was published. In addition, Susan Dart's article "Concepts in Configuration Management Systems" was published in *Proceedings of SCM3*.

The article entitled "Evolving Toward Federated Service-Based CASE Environments" was submitted for publication in *IEEE Software*. Project members also submitted an article entitled "A Model of Tool and Environment Integration" for publication and presentation at the Second International Symposium on Environments and Tools for Ada.

This quarter, project members presented a tutorial at the 13th International Conference on Software Engineering (ICSE-13) entitled "State-of-the-Art in Environment Support for Configuration Management."

Project members made presentations at a sponsor site on the state of commercial configuration management support. Project members also participated in a workshop on configuration management at a sponsor site.

During this quarter, several project members participated in the second meeting of the Navy Next Generation Computer Resources (NGCR) Project Support Environment Standards Working Group (PSESWG) effort. This is an effort undertaken by the Navy to identify and select commercial standards in the environments area for use by the services. The project leader is a member of the executive committee. In a white paper draft, the project leader proposed a reference model of project support environment services. The model takes into account the European Computer Manufacturer's Association/National Institutes of Standards and Technology environment framework reference model. This proposal was accepted by the reference model subgroup as the starting point for a PSESWG reference model.

●●●●●●●●●●●●●●

Domain Analysis

The Domain Analysis Project is developing and testing methods for performing domain analysis to support software reuse. The SEI has reviewed the Domain Analysis project plan for 1991, and has approved the plan subject to several minor changes. Project members are currently addressing issues raised during the review, including one about customers for domain analysis products. Project members met with staff members of the Future Battle Laboratory, Ft. Leavenworth, Kansas; they are interested in using the domain model of movement control as the basis for prototype development. The project will continue to work with this organization and with CECOM.

The Domain Analysis Project is currently analyzing the Army movement control domain to discover and exploit commonality. In performing this analysis, the project is using the Feature-Oriented Domain Analysis (FODA) method, developed by the project in 1990. The analysis provides a model of applications within the domain that can be used to understand the design of those applications and to support the development of new movement control software. The analysis will also serve as a means of improving the FODA method.

During this quarter, the project completed a report defining the context for the domain analysis and received customer approval for its publication. This report documents the scope of the analysis and the relationship of the domain to current Army command and control systems. The report will be published as an SEI special report.

The project has completed the first draft of the domain model of the Army movement control domain. The model currently consists of:

- A definition of the entities related to movement control.

- The operational, mission, and representation features of movement control.

- A high-level functional model, defining movement-control states and activities.

- Detailed documentation of the entities, features and functional models in online and printed format.

To support refinement of the domain model, Jay Stanley, the Army resident affiliate on the project, completed a two-week training course at the Army Transportation School, Ft. Eustus, Virginia. The training concentrated on movement planning and control through various Army echelons.

Project members have presented the draft domain model to CECOM and to combat developers at the Combined Arms Command, Ft. Leavenworth, and received favorable feedback on the ability of the model to abstract commonality within the domain. To discuss the model in depth, three representatives from the Army Combined Arms Support Command (CASCOM) were at the SEI for a week during this quarter. Project members obtained a good idea of ways in which the model can be applied to support CASCOM development efforts. The project plans to hold additional exchanges to refine the model and propose methods for applying it to address CASCOM needs.

The project is also planning a Movement Control Workshop to take place in the third quarter. The workshop will bring together government, industry, and university groups involved in specifying, implementing, or using movement control

systems to support Army operations. The goals of the workshop are to provide a common view of the capabilities of movement control systems and a means by which to represent and exploit that commonality.

A special report, *A Domain Analysis Bibliography* (CMU/SEI-90-SR-3), was produced and appears as an appendix to an Institute of Electrical and Electronic Engineers (IEEE) tutorial entitled "Domain Analysis and Software Systems Modeling."

●●●●●●●●●●●●●●
Domain-Specific Software Architectures

The Domain-Specific Software Architecture (DSSA) Project champions the use of structural models, and the maturation of a structural modeling technology base that embraces all aspects of software development using structural models. As stated by Stewart E. Cranston, Brigadier General, USAF, in a memo to the Secretary of the Air Force/Inspector General: "... a structural modeling methodology has been employed for the Ada Simulator Validation Program, the C-17 Aircrew Training System, the Special Operations Forces Aircrew Training System, as well as the B-2 simulator program managed by the B-2 System Program Office. Structural modeling establishes uniform design implementation across subsystems, providing a rigorous design structure to ensure that the software system has a consistent architecture and is able to meet the requirements. This methodology originated with the ASVP Program in 1987, and it was developed with active participation of Aeronautical Systems Division engineering and the Software Engineering Institute."

During this quarter, project members continued transition of the SOF ATS program to structural modeling. DSSA, the Aeronautical Systems Division, Deputy for Simulators (ASD/YW), and the SOF ATS contractors collaborated to continue evolution of the structural model and structural modeling practices. The results of the collaboration will be showcased at the SOF ATS preliminary design review in July. The SOF ATS is also providing opportunities to extend the applicability of structural modeling to non-real-time systems, including the instructor operator station and the database generation systems.

This quarter, the structural model for the BSY-2 Basic Operator Station gelled. One result is that Naval Sea Combat Systems Engineering Station (NSCSES) has begun to define specific model-based development practices advocated in general by the DSSA. The DSSA will continue to support NSCSES definition efforts, and also will begin work on a change management report describing the impact of structural modeling on development practices.

Project members began planning for the next phase of maturation. In fiscal year 1992, the project will begin a major effort to document the state of the practice of structural modeling and a vision of subsequent community-based maturation. This effort will culminate in an Air Force Handbook on structural modeling for real-time simulation.

●●●●●●●●●●●●●
CASE Technology

The CASE (computer-aided software engineering) Technology Project focuses on improving the ability of SEI sponsors and affiliates to make informed decisions about tool adoption and to improve their practice in the use of CASE tools. It

also provides information to tool vendors on current tool usage and gaps in current technology.

Project members continued the study of "lessons learned" from CASE adoption experiences, which has been analyzing how actual projects have used CASE tools. This effort has provided input to the *Guide to CASE Adoption*, which is scheduled to be published early in the fourth quarter.

During this quarter, project members collaborated closely with the Software Development Environments Project in addressing the issue of tool integration. This SEI study, completed for the NGCR Committee, assessed the use of current standards in the building of software engineering environments, and identified the need for additional standards. This joint work has helped to identify new trends in cooperative interactions between tools.

Project members are continuing to design the survey of the experiences of CASE tool users, in conjunction with the SEI Empirical Methods group. The survey will address such issues as readiness for CASE tools, the adoption process, tool requirements and needs, and tool issues and problems.

A workshop on CASE management issues was held in June. The issues included CASE acquisition, CASE selection, CASE and metrics, the real impacts of CASE, and CASE readiness. Representatives from industry and government participated in the exchange. The periodic workshops consider critical issues in the CASE area and provide a forum for vendors and users to address underlying issues in an objective setting.

●●●●●●●●●●●●●
**Requirements
Engineering**

Although there have been substantial advancements in software engineering methods and tools during the past twenty years, requirements engineering still remains a key problem area in the development of complex, software-intensive systems. Issues involved in this problem area include:

- Long requirements development time.
- Analysis and validation difficulty.
- Frequent requirements changes.
- Unmanageable documentation.

The Software Engineering Techniques Program initiated the Requirements Engineering Project during the first quarter to address these issues. As its first task, the project held a workshop in March to gain insight into these requirements issues, to understand the problems customer organizations have, and to identify their needs. Some of the recommendations from the workshop were to:

- Adopt an evolutionary, incremental approach.
- Adopt a multi-disciplinary team approach.
- Do more up-front requirements work using prototyping.
- Separate user interface from the functional requirements.
- Capture major decisions and rationales.

Based on these recommendations, the project has been developing a detailed plan focusing on requirements engineering processes, prototyping techniques, formal methods and requirements elicitation techniques, and the integration of these elements into a requirements engineering handbook. The project presented the plan at a CECOM meeting in June and is currently performing several informal reviews to improve the plan.

A Domain Analysis Bibliography
(CMU-SEI-90-SR-3)

Tool Integration and Environment Architectures
(SEI-91-TR-11)

Case Studies in Environment Integration
(SEI-91-TR-13)

◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊

Software Engineering Techniques Reports

April–June 1991

For information on ordering copies of SEI reports, see page 37.

Real-Time Distributed Systems

The goal of the Real-Time Distributed Systems Program is to ensure that the developers of all major real-time distributed software systems in the DoD routinely employ quantitative methods to evaluate software engineering designs, upgrades, and tradeoffs.

The Rate Monotonic Analysis for Real-Time Systems Project aims to ensure that rate monotonic analysis and scheduling algorithms become part of the standard practice for designing, building, troubleshooting, and maintaining real-time systems. Rate monotonic analysis (RMA) helps engineers to understand and predict the timing behavior of hard real-time systems to a degree not previously possible.

●●●●●●●●●●●●●●
Rate Monotonic Analysis for Real-Time Systems

Goals of the project include increasing the use of rate monotonic theory on highly visible projects, obtaining support for the rate monotonic approach from national hardware and software standards, and promoting the development of training and consulting resources outside the SEI. These resources are intended to serve as the major means for widely transitioning RMA techniques to real-time system developers.

As part of the goal to increase the use of rate monotonic theory on highly visible projects, project members are working closely with the prime contractor for the Navy's BSY-2 system, a major distributed system, to ensure the successful use of RMA.

During this quarter, project members worked with various BSY-2 software design engineers and managers in applying rate monotonic analysis to their software designs. Data sheets developed previously by the project were used to collect information needed to perform the analysis. The analysis determined whether a particular set of tasks would meet its timing requirements both when operating independently on a particular node and when integrated with other sets of tasks. For some task sets, the results of the analysis showed that performance could be improved significantly with minimal impact on the overall software design. Project members continue to work with BSY-2 personnel to complete the analysis for the integration of particular task sets. Project members have nearly completed a report that explains how to perform RMA on BSY-2 software designs; this report will be disseminated throughout BSY-2.

Project members are working with the Navy's NGCR Program to encourage the development of a local area network (LAN) that adequately supports rate monotonic principles. Project members have continued to develop an Ada binding to the SAFENET Lightweight Application Service definition. The target implementation is the Xpress Transport Protocol developed by Protocol Engines Inc. Project members continue to work on system end-to-end scheduling issues and have also examined the IEEE 802.6 network standard for its applicability to the real-time domain. A CMU Ph.D. candidate has been encouraged to use a proposed real-time extension to IEEE 802.6 as a vehicle for his research, thereby

providing additional resources to develop a real-time LAN design that supports RMA.

The NGCR has selected IEEE Futurebus+ (IEEE 896) as its computer backplane standard. During this quarter, project members continue to support the transition of Futurebus+ to the real-time community. The project has worked to ensure that rate monotonic scheduling is the recommended approach for real-time applications using Futurebus+. At this point, internal balloting on the FutureBus+ system design manual (IEEE 896.3) has been completed. Out of 30 total votes, there were 4 votes against the use of rate monotonic scheduling. Project members have begun working with the committee to revise the manual and deal with the negative votes. In a separate effort, NanoTech, a small hardware company, has announced single board computer products for Futurebus+. The product announcement advertises support for RMA and claims worst case priority inversion of no more than 100 microseconds.

Project members have begun developing Ada Usage/Performance Specifications. The goal of this work is to ensure that if programmers follow certain coding restrictions, they will get the benefit of certain performance improvements in generated code and runtime system behavior. In addition, the Usage/Performance Specifications will help identify issues that should be addressed by runtime system implementors. The first issue examined was delay statement processing. Typically, the expiration of a delay causes the current executing task to be suspended so the task whose delay has expired can be scheduled. This can lead to a form of priority inversion in which a high-priority task may be blocked because the runtime system is providing service to a lower priority task. An alternative approach to treating the expiration of a task delay has been developed and is being examined with regard to performance impact. We have also completed a Memorandum of Understanding with Systems Designers to collaborate in the Ada Usage/Performance Specification work and are working to establish similar memoranda with other compiler vendors.

A team of project members has been interacting with the prime contractors of the Air Force's PAVE PACE program to ensure that they are aware of RMA and to encourage them to use rate monotonic principles when they offer system design proposals to the Air Force. The project is interested in PAVE PACE because the program is defining the next generation architecture for Air Force avionics systems. During this quarter, project members learned that RMA has been incorporated into one of the architectures. The team will be carrying on its work with RMA during the next phase of the project.

During the downsizing of its space station, NASA investigated RMA and concluded that, although the other rate monotonic scheduling options were desirable, only the basic priority inheritance algorithm would be directly supported by the operating system and Ada runtime. As a result, basic priority inheritance was the only funded RMA option. However, project members have been working on RMA with Lynx (the operating system developer) and Alsys (the compiler vendor). As a result, Lynx and Alsys agreed to provide support for the sporadic server algorithm, at their own cost, as part of their commercial productization effort. Lynx will also implement the priority ceiling protocol. After coordinating with NASA, McDonnell Douglas, IBM, Lynx, and Alsys, the Space Station project agreed to use the additional RMA options at no cost to NASA or its prime contractor.

Project members visited Boeing Aerospace from March 25-29 at Boeing's request (via Edith Martin, a member of the SEI Board of Visitors) to determine the schedulability of the software redesign of the dual-processor SRAM-II (Short Range Attack Missile) system written in Ada and assembly language. Rate monotonic analysis was applied to the free flight software mode in both processors. This mode was of concern to Boeing because the resulting high processor utilizations may cause deadlines to be missed. Based on the detailed designs and timing information, the rate monotonic analysis verified that in both processors all deadlines would be met in free flight mode. The analysis technique and results were explained in detail to the lead design engineer, who was pleased with the results and impressed with the techniques, and who would like to continue the analysis throughout the remaining life cycle of the SRAM-II project.

Project members continue to work toward the goal of producing an engineering handbook, exploring two key ideas for its organization. The first idea is to develop the handbook around common real-time problems (or patterns). The basic idea is to view real-time systems as a collection of real-time patterns and provide a description of the rate monotonic analysis associated with each pattern. To test this idea, project members wrote sample problem descriptions. The second idea is to include in the engineering handbook a taxonomy of issues to be addressed in designing real-time systems from which a comprehensive list of real-time patterns could be derived. A first draft of the taxonomy has been written and is currently being reviewed.

The RMA Adoption Rationale (the question and answer report) has been reviewed by external and internal reviewers, and substantive comments were received. All reviewers felt that this type of report would be very useful.

Project members continued to work with the Advanced Real-Time Technology Project at CMU to extend the current analytical methods for assessing schedulability. The results obtained allow task sets to be analyzed where each task's execution priority may vary during different intervals of computation. These results were documented in a report that has been submitted for presentation at the next IEEE Real-Time Systems Symposium.

Project members wrote a technical note describing several algorithms for assessing the schedulability of a task set with arbitrary deadlines. The algorithms are potentially much more efficient than the one used in a prototype schedulability analysis tool that was previously written by the project.

The Real-Time Embedded Systems Testbed (REST) Project is collecting, classifying, generating, and disseminating information about Ada performance issues in real-time embedded systems.

Real-Time Embedded Systems Testbed

Building real-time systems, particularly embedded and distributed real-time systems, is currently a craft rather than a structured, methodical endeavor. Tools, including the Ada programming language and the accompanying Ada runtime support facilities, are only beginning to fulfill their promise for raising the level of abstraction for real-time programming. Use of such tools is still perceived as entailing a high degree of risk. One of the problems addressed by REST is the identification and the reduction of such risks.

At this time, the government does not routinely assess the quality of Ada compilation systems. Both Ada users and vendors need help to improve the quality of such systems. The identification, development, and distribution of techniques and tools to help improve Ada implementations is another problem addressed by REST.

This quarter, the project continued to refine the detailed design of the Hard Real-Time (Hartstone) Benchmark Suite. The first benchmark of the suite, the periodic harmonic tests, is currently being used by nearly 300 sites. The European Space Research and Technology Center, an organization of the European Space Agency, has used Hartstone to evaluate Ada compilers for space applications. Several vendors are now routinely using Hartstone to evaluate their products, and some C versions of Hartstone that use the UNIX process model have appeared. Research continued at CMU on a distributed version of the benchmark, and MITRE modified Hartstone in support of a major procurement for NASA Johnson Space Center.

In support of BSY-2, the project is investigating the performance impact of Ada style using Ada Compiler Evaluation Capability (ACEC), Ada Evaluation System (AES), Performance Issues Working Group (PIWG), and Hartstone benchmarks. Project members are participating in monthly meetings with General Electric and the compiler vendor, Verdix, to continue improving the compilation system and the performance of the BSY-2 software. A first draft of a document critiquing the BSY-2 Ada style guide from a performance standpoint was offered for review.

The use of the existing evaluation technology such as ACEC, PIWG, and AES on a major project such as BSY-2 led to recommendations for improvement that were forwarded to the sponsors and the developers of this technology. Such feedback is timely in light of the recent agreement between the U.S. and UK governments to merge their respective evaluation suites (ACEC and AES). A workshop is being planned at the SEI to discuss the policies that could be implemented to foster the use of the combined evaluation system.

The project developed a half-day tutorial on evaluation and benchmarking that summarizes years of experience of the REST Project. The tutorial was offered at the Washington Ada Symposium and increased the awareness of government and industry representatives.

Cooperation with several companies (IBM, WEC, Boeing, SPC, and MITRE) and numerous conversations with vendors and users of Ada compilation systems contributed to a better awareness of evaluation and benchmarking issues on a broad range of projects such as the Space Station Freedom Program (NASA), the Advanced Automation System, the Radar Software Improvement Program and Short Range Attack Missile (Air Force).

Project members increased their collaboration with the Comet Rendezvous Asteroid Flyby (CRAF) and Cassini mission of NASA's Jet Propulsion Laboratory (JPL). They are following the progress of and offering occasional advice to the mission's development team, which is designing a new generation of spacecraft. The space-craft are scheduled to study Saturn and its moons (Cassini mission) and the smaller bodies of the solar system (CRAF mission). The tremendous radiation on Jupiter and Saturn requires MIL-STD-1750A hardware architecture; the flight soft-ware is being developed in Ada. Project members organized a workshop about MIL-STD-1750A and Ada at JPL, which was attended by nearly 100 people and

all vendors of Ada/1750A compilers. The workshop fostered a healthy exchange of technical data and lessons learned between vendors, users, and policy makers.

Together with the Rate Monotonic Analysis for Real-Time Systems Project, REST is investigating classes of benchmarks that would produce the parameter values necessary to perform an accurate rate monotonic analysis of the schedulability of Ada tasks. Project members also began evaluating the overhead of sporadic servers. With the Distributed Systems Project, including a resident affiliate from Hughes Aircraft Company, REST is investigating how Specification Methodology for Adaptive Real-Time Systems (SMARTS) could be applied to the automatic generation of application-specific benchmarks.

●●●●●●●●●●●●●●
Distributed Systems

The Distributed Systems Project (formerly Software for Heterogeneous Machines) is developing tools and a methodology for building distributed, large-grained, concurrent applications to run on networks of heterogeneous machines. The project has developed Durra, a language for describing distributed applications as a set of task descriptions and for writing type declarations that prescribe a way to manage the resources of a network.

The REST and Distributed Systems Projects completed a Memorandum of Understanding describing a number of joint activities. Planned activities for 1991 include porting Durra to the REST testbed for performance evaluation, fine-tuning and using Durra to configure and build distributed benchmarks, and evaluating SMARTS as a benchmark description language. Plans for 1992 include automatic generation of benchmarks from formal specifications. (Generating benchmarks is a problem of interest to the project, and members are developing techniques for application prototyping and automatic program generation.)

Project members are having difficulty using the Verdix Ada Development System (VADS) for the Sun4. This system was recently acquired and installed on the project computers, and execution errors in Durra test cases are being discovered. These errors are difficult to debug because they destroy runtime system information, and project members do not have access to VADS sources. To date, the cause of the problem is unknown. Support for the Sun4 version of VADS was transferred from VERDIX to Sun as of early June, and Sun has not yet developed a customer support mechanism for their new product (SunAda).

Problems using VADS are delaying the development of the new Durra runtime. Project members continue making progress with the Durra compiler and cluster generator.

Project members attended a meeting of the IEEE 1003.5 (Ada language binding to 1003.1, POSIX System API) Working Group in Chicago on April 15-19.

A paper written by project members was accepted for TriAda, which takes place in San Jose, California, on October 21-25, 1991.

Project members presented project activities at the IEEE 2nd International Workshop on Rapid System Prototyping (June 11-13, Research Triangle Park, North Carolina); the Department of Computer Science, Imperial College (June 10, London, England); and the 3rd International Software Configuration Management Workshop (June 12-14, Trondheim, Norway).

User Interface

The User Interface Project has developed Serpent, a user interface management system (UIMS). Serpent separates the concerns of the user interface from those of the application, which allows integration of input/output technologies without modification of the functional portion of the application.

Serpent is distributed electronically through anonymous file transfer protocol (FTP) and is supported electronically through a mailing list. Currently, more than 150 Serpent sites from research, academia, and industry are on the mailing list.

During this quarter, the final version of Serpent was released. Version 1.0 included an alpha version of the dialogue editor, a completed Ada interface, and completely revised documentation.

An announcement about the opportunity to license Serpent from CMU will appear in *Commerce Business Daily* in early July. Following this announcement, interested parties will have a specified period of time to submit proposals about their intentions to market Serpent.

The User Interface Developers Workshop, organized by the project, held two meetings this quarter. A white paper produced by the workshop that presents a reference model for future user interface tools was presented to the Conference on Human Computer Interaction in May. Papers that cite the white paper are already beginning to appear in the literature.

The project is terminating as of July 1. The goals of the project were to make UIMS technology generally available, and the project is aware of two major vendors and several minor ones that plan to offer UIMS products in the next several years.

Systems Fault Tolerance

The Systems Fault Tolerance Project was initiated to promote the use of fault tolerance in the implementation of dependable safety-critical systems.

The Systems Fault Tolerance Project hosted a Fault Tolerant Systems Practitioners Workshop at the SEI in June. Individual practitioners from Hughes Aircraft (Ground Systems Group and Radar Systems Group), IBM Federal Sector Division (Space Shuttle and Air Traffic Control), Honeywell Air Transport Systems, the Aerospace Corporation, Martin Marietta Astronautics, Stanford Research International, and Allied-Signal Aerospace attended this workshop. Collectively, they are involved in the development of fault tolerant systems for applications ranging from space to avionics to traffic control. The workshop discussed the state of the practice in the application of fault tolerance to deployed systems, barriers to the more effective use of fault tolerance in system implementation, and fault tolerance technology needs. A summary report is in preparation.

A project member attended the most recent meeting of the IEEE Software Safety Plan Standards Committee. This committee is developing a proposed standard (IEEE Standard P1228) on the specification of software safety plans. Safety-critical systems are a major application for fault tolerance technology.

Serpent Overview
(CMU/SEI-91-UG-1)

Serpent: System Guide
(CMU/SEI-91-UG-2)

Serpent: Saddle User's Guide
(CMU/SEI-91-UG-3)

Tool Integration and Environment Architectures
(CMU/SEI-91-UG-5)

Serpent: Guide to Adding Toolkits
(CMU/SEI-91-UG-6)

Serpent: C Application Developer's Guide
(CMU/SEI-91-UG-8)

For information on ordering copies of SEI reports, see page 37.

◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊

Real-Time Distributed Systems Reports

April–June 1991

Software Risk Management

The goal of the Software Risk Management Program is to improve the management of risks in the development of software-intensive systems. In this context, "risk" refers to the uncertainty and impact associated with an event; "management" refers to the identification and resolution of the risk. Managing risk, therefore, entails identifying those things that can go wrong and assessing their likelihood and impact. A premise of the Risk Program is that confronting risk in a systematic way is fundamental to controlling the quality, cost, and schedule of software products.

Lessons from the first industry assessment have been incorporated in our assessment process. These changes involve more extensive interaction with our client's program management chain prior to the actual on-site period (the two-step process for performing the risk assessment that was used on-site was extremely effective) and finding more effective ways to communicate risk information. Work is continuing on the second industry assessment, which is expected to occur during the third quarter.

The Risk Program is continuing to conduct interviews with industry to gather information on the current practice of risk management and risk communication.

The program has developed a questionnaire for the government (derived from the Risk industry questionnaire) that will gather information on how DoD services handle risk management and risk communication. The questionnaire was used in Risk's first government program office interview in May 1991.

Insights gained from the first industry assessment are also being factored into the risk identification mechanisms. In addition, complete documentation packages for the mechanisms are being developed in preparation for the next industry assessment.

Project members have initiated a data analysis activity. All of the data collected from interviews have been assembled into a preliminary database. Project members plan to refine the database and incorporate risk assessment data during the third quarter. The first analysis and output from this database will be a state-of-the-practice report on software risk management, scheduled for completion during the third quarter.

Another new activity is the development of a risk analysis capability. The assessment clearly demonstrated that risks can be identified. Risk analysis is the conversion of risk data into risk management information; that is, information needed by decision makers to take action on risks. The risk analysis work will greatly enhance the project's support of the risk management process.

Project members are following up with each of the attendees of the government workshop held at the Defense Systems Management College (DSMC). As a result of this follow-up, project members have opened TO&P discussions with a Program

Executive Officer (PEO) to discuss the details of establishing a strategic partnership. Once established, this partnership will allow the project to develop its methods and processes and apply them to PEO program offices and contractors.

Project members are continuing to work with the National Security Industrial Association (NSIA) on the risk conference for tne fall. The Call for Participation for the Software Risk Management Conference has been distributed to all NSIA members and SEI affiliates. The conference will be held at the SEI in October.

Work has begun on the proposal for the Software Risk Management course to be taught as part of the SEI Continuing Education Executive Series. The draft Education and Training Review Board (ETRB) proposal for the course is currently undergoing program review. A review by the full ETRB is planned for the third quarter.

Work has begun on the development of risk profiles. Risk profiles are another vehicle, similar to risk assessments, for identifying risks. In addition, risk profiles will emphasize risk communication and mitigation strategies.

Special Projects

Staff members of the Transition Models Project will develop a set of methods for planning, implementing, and assessing transition activities; these methods will be useful for technology producers and consumers both inside and outside the SEI. Transition Models staff will also provide SEI staff with education and training on technology transition concepts and approaches.

The project leader continued coordinating SEI support and providing input on technology transition to the effort to develop the DoD Software Technology Plan. In this quarter, the project leader attended planning meetings and working sessions, revised the draft of the technology transition section of the plan, and submitted the section to the coordinators of the plan at the Office of the Secretary of Defense. Frank Sisti will assume coordination responsibility in July and will provide full-time support to this work.

Several project members presented a tutorial, "Software Technology Transition," at the 13th International Conference on Software Engineering on May 12 in Austin, Texas.

Project members planned and facilitated the STARS Vision Workshop with the STARS Program Director and Program Architects on April 14-18. They also facilitated a continuation of the workshop on May 16-17.

At the request of the STARS Joint Architects Team, Stan Przybylinski reviewed the previous issues of the STARS Program newsletter to provide constructive feedback on how to improve the publication in subsequent issues. The suggested changes were implemented, and readers and STARS management expressed satisfaction with the improvement.

Project members are planning a workshop on technology transition in coordination with a representative from Electrical Power Research Institute. Workshop attendees will consist of those involved in technology transition activities at organizations represented on the Council of Consortia CEOs.

Project members met with Nilton de Oliveira to discuss the technology transition strategies of TELEBRAS.

The project initiation review for the Transition Models Project was held on June 11. Useful feedback was received, which will be incorporated into a revised project plan.

The Empirical Methods group supports technology development, assessment efforts, and SEI technology projects by providing market research methods and materials, conducting surveys, and evaluating events or validating products of SEI projects. In addition, the group conducts the National Software Capacity Study.

During the second quarter, Empirical Methods staff and colleagues from the Carnegie Mellon School of Urban and Public Affairs presented interim results of the National Software Capacity Study to senior Air Force and Army officials. The

interim results show that in spite of major program cancellations and reduced defense spending, demand for systems built with Ada increased from 1989-1991 by about 12 million source lines of code. The post-deployment software support (PDSS) load for organic work for the Air Force alone is now at about 33,000 computer program configuration items. Additional software being maintained by two Air Force commands that provided information to the study team is approximately 50 million source lines of code. Recent estimates show the cost to fix a line of code during initial development in the U.S. is $50, while the cost to fix it in PDSS is $4000.

The interim briefing also indicates that as software demand is increasing in the U.S., the supply of engineering talent to produce software-intensive systems is declining. Furthermore, the resources to develop people with the requisite education and training are not in place. U.S. software producers presently hold the global lead (about 55% of the world's $110 billion software development market). However, a substantial proportion of software production is moving to Asia (Japan, Singapore, India), Western Europe, and Canada. Interim initiatives are urgently needed to consider the need for federal legislation. Also, consideration should be given to improving coordination among key agencies (e.g., the DoD, Department of Education, National Science Foundation, Office of Personnel Management) to plan and implement appropriate actions to address the supply problem and the risk of losing the software industry to offshore competition.

Software Engineering Education

The primary objective of the Software Engineering Education Program is to increase the number of highly qualified software engineers by rapidly improving software engineering education throughout the education communities of academia, government, and industry. To accomplish this, the projects of the Education Program focus on accelerating the development of software engineering programs in academia and on enhancing opportunities for the continuing education of practitioners.

The Software Engineering Curriculum Project recently changed its name to Graduate Curriculum Project, and a new Undergraduate Curriculum Project has been established. Focusing on the appropriate academic level, each project is developing model curricula, promoting the growth of software engineering programs in the academic community, and working to increase the amount of software engineering content in computer science programs. The Graduate Curriculum Project produces educational materials, including the Academic Series, a set of graduate-level videotaped courses on software engineering. The materials developed by the Undergraduate Curriculum Project will include appropriate undergraduate-level prerequisite materials for the courses developed by the other projects in the Education Program.

● ● ● ● ● ● ● ● ● ● ● ● ● ●
Software Engineering Curriculum

James Tomayko, the conference chair, submitted the final text for the proceedings of the Fifth Conference on Software Engineering Education to the publisher, Springer-Verlag. More than two dozen papers will be presented at the two-day conference, which will be held on October 7-8, 1991 in Pittsburgh. The conference will be hosted by the SEI and will be held in cooperation with the Association for Computing Machinery (ACM) and the IEEE Computer Society.

The Academic Series ended its eighth semester of production by completing the videotaping of three courses: "Software Creation and Maintenance" (version 2); "Software Project Management" (version 2); and "Software Analysis" (updates Software Verification and Validation). Presentation of these courses began at Carnegie Mellon in January; the videotaped versions will be available to participating schools for the fall 1991 semester.

The project completed a package of educational materials: *Scenes of Software Inspections: Video Dramatizations for the Classroom* (CMU/SEI-91-EM-5). The package includes a videotape and a report. The tape contains scenes of appropriate and inappropriate behavior at software inspections and examples of other types of group interaction. The report includes a description of each scene, along with suggestions for using the videotape in the classroom.

The *Software Engineering Education Directory* (CMU/SEI-91-TR-9) was released this quarter. The directory provides information about software engineering courses and degree programs offered by universities, primarily in the United States.

•••••••••••••••
Master of Software Engineering

In response to industry's growing demand for skilled software developers, CMU offers a 16-month master's degree program in software engineering. The program is a joint effort of the Carnegie Mellon School of Computer Science and the SEI. The core of the program is based on the SEI curriculum recommendations for MSE programs.

The 1991 summer session began on May 28. In the Software Development Studio, an ongoing project course, students are completing a software product—a system that allows users to graphically manipulate database objects—for July 26 release to the customer. In conjunction with the studio, students began the Software Engineering Seminar, which focuses on communication skills and interaction with well-known software professionals. Both courses will continue in the fall semester.

•••••••••••••
Continuing Education

The Continuing Education Project interacts with industry and government to increase the availability of high-quality educational opportunities in software engineering topics for software practitioners and executives. The project produces the Continuing Education Series and the Technology Series. The Continuing Education Series offers two types of courses: video-based courses designed for clients' in-house education, and executive offerings for decision makers involved in improvement efforts. The Technology Series provides stand-alone presentations that promote awareness of emerging issues and leading-edge technologies.

This quarter, the project presented two executive seminars from the Continuing Education Series. On April 15-16, 19 people attended "Software: Profit through Process Improvement." They represented Defense Information Systems Agency, Electronic Systems Division – Hanscom AFB, Grumman Melbourne Systems, Honeywell, IBM, Schlumberger, Scott AFB, Sterling Software, and the U.S. Navy.

On May 6-8, "Software Quality Improvement" was presented to 26 managers from Airspace Technology, Ashton-Tate, BDM International, General Dynamics, Honeywell, Lawrence Livermore National Laboratory, Martin Marietta, Naval Surface Warfare Center, OAO Corporation, Pacific Bell, PM Trade, Telos Systems Group, CECOM, and the U.S. Navy.

At Westinghouse Electric Corporation, SEI staff participated in an in-house offering of the Continuing Education Series course, "Software Project Management." The staff delivered lectures to 20 students at the Westinghouse facility in Monroeville.

Project members are developing two new courses in the Continuing Education Series: "Software Productivity Improvement," an executive seminar, and "Software Requirements Engineering," a video-based course for organizations' in-house use. The Education and Training Review Board reviewed a proposal for "Software Productivity Improvement," and the design review of "Software Requirements Engineering" is scheduled for July.

As part of work defined by the Air Force Computer Resource Management Technology Program, project staff developed a prototype instrument called the Software Engineering Training Needs Analysis Survey. On April 23, project members admin-

istered the survey at the first pilot site, Electronic Security Command in San Antonio, Texas. The instrument has been developed for organizations using DoD development standards.

1991 SEI Report on Graduate Software Engineering Education
(CMU/SEI-91-TR-2)

Software Engineering Education Directory
(CMU/SEI-91-TR-9)

Scenes of Software Inspections: Video Dramatizations for the Classroom
(CMU/SEI-91-EM-5)

For information on ordering copies of SEI reports, see page 37.

◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊·

Software Engineering Education Reports

April–June 1991

DARPA/STARS Support

As Ada use becomes more common in software-dependent systems, and as requirements grow, system developers will be faced with both the benefits and problems of adopting more disciplined and integrated approaches to software engineering. The goals of DARPA/STARS Support are to serve as a focal point and facilitator in removing technical and managerial impediments to the adoption of Ada; to support the STARS Program in technology development, integration, and transition efforts; and to support DARPA efforts to develop, evaluate, and disseminate new software engineering approaches derived from the development of software architectures characteristic of particular application areas.

●●●●●●●●●●●●●●
Binding of Ada and SQL

The Binding of Ada and SQL Project, initiated at the request of the Ada Joint Program Office, has investigated the problem of binding the Ada programming language with the Structured Query Language (SQL) database language. The solution to this problem was the specification of the SQL Ada Module Extension (SAME), an interface that permits an application program written in Ada to access and manipulate data controlled by a database management system (DBMS) using SQL.

This quarter, the project leader participated in a panel session at the Summer SIGAda meeting held in McLean, Virginia, on June 18.

The document *Rationale for SQL Ada Module Description Language SAMeDL* was published as an SEI technical report, CMU/SEI-91-TR-4, during this quarter.

Also during this quarter, the project leader presented a paper at the Database Colloquium in San Diego, California, on June 25.

The SAMeDL language specification was approved as a Committee Draft by the ANSI database committee (X3H2) and by the US Ada Technical Advisory Group (TAG).

◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊
DARPA/STARS Support Reports

April–June 1991

Rationale for SQL Ada Module Description Language SAMeDL
(CMU/SEI-91-TR-4)

For information on ordering copies of SEI reports, see page 37.

CERT Coordination Center

The Computer Emergency Response Team Coordination Center (CERT/CC) supplements existing mechanisms through which informally organized experts deal with and prevent computer emergencies. The CERT/CC at the SEI supports two different communities: Internet users and developers of technology that is available on the network, such as UNIX and networking software. The CERT/CC provides a dependable 24-hour point of contact for security issues and allows rapid communication during emergencies. It also raises constituents' awareness of security issues and assists individual organizations in improving the security of their systems. Finally, the CERT/CC maintains a highly secure repository of information for team members and cultivates close ties with researchers in the area of trusted systems to improve the security of existing systems.

Since its inception in 1988, the CERT/CC has responded to a continuous stream of reported security incidents. These include reports of intrusions, worms, and viruses as well as reports of vulnerabilities and suggested fixes for problems. In handling these problems, the CERT/CC issues advisories to the Internet community to warn them of problems and inform them of preventive techniques. In cases where vulnerabilities exist, the CERT/CC works with software vendors and the technical community in analyzing and resolving the problems.

During the second quarter, the CERT/CC continued working with the Internet Engineering Task Force (IETF) to produce guidelines for the secure operation of the Internet. The guidelines are currently available as Internet drafts, and the Internet Engineering Steering Group has recommended that the guidelines be approved as a final Informational RFC (request for comments). (Final RFCs have been reviewed and approved, and are not open for further comment.)

CERT members participated in the Site Security Policy Handbook Working Group, a second IETF working group whose aim is to produce a security handbook for use by Internet-connected site and system administrators. Ongoing meetings of the working group have resulted in a second draft of the handbook, which outlines key issues and provides guidance on policy, administrative, and technical issues to support administrators' efforts at improving the security of their systems. Final editing of the handbook has been completed, and it is available for comment as a draft RFC.

The CERT staff has been beta testing Trusted Information Systems' Privacy Enhanced Mail as part of a larger effort to evaluate the package and feasibility of use on a wide scale. The package uses private and public key encryption techniques to authenticate mail senders and to protect the privacy of the

transmitted message. Initial testing has been promising, and plans call for expanded testing over the next several months.

The first draft of a computer security tutorial, focused on Internet-connected UNIX systems, was completed. The tutorial will be presented at the Computer Security Incident Response Workshop in August.

CERT/CC staff members hosted working meetings and participated in professional conferences. The focus of these activities is to raise awareness of security issues and to support organizations' efforts at improving the security of their operational systems.

The CERT/CC continued to take a lead position in the development of the CERT System, a federation of organizations working together to improve the security of their systems. The program for the third annual Computer Security Incident Response Workshop has been completed. This workshop will be held in Washington, D.C., on August 6-8, 1991. Papers, invited presentations, and workshops will cover the following topics: network intrusions, incident handling procedures and policies, legal and investigative issues, system vendor activities, ethical issues, vulnerabilities and malicious code, security of wide area networks, and CERT system operations.

Affiliate Relations

The Affiliate Relations Function establishes and maintains SEI relationships with industry, government, and academia by providing access to SEI information through mailings, telephone contact, special meetings, and symposia. Affiliate Relations also cooperates with the SEI Joint Program Office to negotiate for and place resident affiliates at the SEI.

New industry affiliates from five organizations signed information exchange agreements during this quarter: Cray Research, Inc.; Dynamics Research Corporation; McDermott, Inc.; Research Triangle Institute; and Spectrum International, Inc.

New academic affiliates from seven institutions also signed information exchange agreements during this quarter: Colorado State University, Dublin City University, Monmouth College, Ohio State University, Texas Christian University, University of Scranton, and University of Southwestern Louisiana.

New resident affiliates, one from Computer Sciences Corporation and one from the DoD, recently joined the SEI. As of June 30, 13 resident affiliates were working at the SEI: 7 from industry, 1 from academia, and 5 from government agencies and the services. In this quarter, 2 resident affiliates concluded their technical project assignments at the SEI.

Affiliate Relations conducted SEI Visitors Day on June 20; 62 representatives from industry, government, and academic organizations were in attendance. Special meetings were held later in the day for visitors from AT&T Bell Laboratories, Raytheon Company, and Sematech. The next Visitors Day is scheduled to take place on November 13, 1991. For additional information about Visitors Day, contact Affiliate Relations (see page 37).

For More Information

To order an SEI publication, send a written request accompanied by a pre-addressed mailing label to:

> Software Engineering Institute
> ATTN: Publications Requests
> Carnegie Mellon University
> Pittsburgh, PA 15213-3890

**How to Order
SEI Publications**

To be added to the Bridge mailing list, write to:

> Software Engineering Institute
> ATTN: Bridge Mailing List
> Carnegie Mellon University
> Pittsburgh, PA 15213-3890

**How to Get the
SEI Magazine**

For information on opportunities for affiliation, contact:

> Software Engineering Institute
> ATTN: Mark Coticchia
> Carnegie Mellon University
> Pittsburgh, PA 15213-3890
> (412) 268-6138
> Internet: mec@sei.cmu.edu

**How to Become
an SEI Affiliate**

For further information about the SEI, contact:

> Message Center
> Software Engineering Institute
> Carnegie Mellon University
> Pittsburgh, PA 15213-3890
> (412) 268-7700

**How to Get
Additional Information**

List of Acronyms

ACEC   Ada Compiler Evaluation Capability
ACM   Association for Computing Machinery
AES   Ada Evaluation System
AFECO   Air Force Electronic Combat Office
ASD/YW   Aeronautical Systems Division, Deputy for Simulators
CASCOM   Combined Arms Support Command
CASE   Computer-Aided Software Engineering
CERT/CC   Computer Emergency Response Team Coordination Center
CRAF   Comet Rendezvous Asteroid Flyby
CSITE   Combat System Integrated Training Equipment
DARPA   Defense Advanced Research Program Agency
DoD   Department of Defense
DSMC   Defense Systems Management College
DSSA   Domain-Specific Software Architecture
ESD   Electronic Systems Division
ETRB   Education and Training Review Board
FODA   Feature-Oriented Domain Analysis
FTP   File transfer protocol
Hartstone Hard Real-Time Benchmark Suite
ICSE   International Conference on Software Engineering
IEEE   Institute of Electrical and Electronic Engineers
IETF   Internet Engineering Task Force
JPL   Jet Propulsion Laboratory
LAN   Local Area Network
NADC   Naval Air Development Center
NCSC   Naval Coastal Systems Center
NGCR   Next Generation Computer Resources
NSCSES   Naval Sea Combat Systems Engineering Station
NSIA   National Security Industrial Association
PDSS   Post-Deployment Software Support
PEO   Program Executive Officer
PIWG   Performance Issues Working Group
PSESWG   Project Support Environment Standards Working Group
RMA   Rate Monotonic Analysis
REST   Real-Time Embedded Systems Testbed

SAE	Software Architectures Engineering
SAME	SQL Ada Module Extension
SCCM	Software Capability Maturity Model
SCE	Software Capability Evaluation
SQL	Structured Query Language
SCM	Software Configuration Management
SDE	Software Development Environments
SPA	Software Process Assessment
SMARTS	Specification Methodology for Adaptive Real-Time System
SRAM	Short Range Attack Missile
SSC	Standard Systems Center
STARS	Software Technology for Adaptable, Reliable Systems
TO&P	Technical Objectives and Plans
UIMS	User Interface Management System
VADS	Verdix Ada Development System

SOFTWARE ENGINEERING INSTITUTE